DOCUMENT RESUME

ED 431 432 IR 057 394

AUTHOR	Chimi, Carl J.; Gordon, Gene M.
TITLE	Issues on the Leading Edge: An Examination of Problems Faced by Innovative Instructors of Information Technology.
PUB DATE	1998-12-00
NOTE	8p.; In: Proceedings of the International Academy for Information Management (IAIM) Annual Conference (13th, Helsinki, Finland, December 11-13, 1998); see IR 057 374.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Budgets; Computer Software; *Computer Uses in Education; Higher Education; Information Science Education; *Information Technology; *Instructional Innovation; Problems; Technological Advancement; World Wide Web
IDENTIFIERS	Computer Users; File Transfer Protocol; *Technology Utilization

ABSTRACT

This paper examines the problems faced by instructors of information technology when they try to introduce new technologies for delivering instructional material. Some of the issues discussed include: (1) dealing with new versions of software when the institution wants to keep older versions on its computers; (2) dealing with budgets, both as constraints on resources and as inefficient mechanisms for providing leading edge technology; (3) dealing with colleagues, staff, and students who are not always in synch with leading edge needs; (4) dealing with students whose computers and software are ahead of instructors'; and (5) rewards and pitfalls of using innovative technologies to deliver course materials, including the use of the World Wide Web for online syllabi and exams and the use of FTP (File Transfer Protocol) for delivering assignments. (Author/MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 431 432

ISSUES ON THE LEADING EDGE: AN EXAMINATION OF PROBLEMS FACED BY INNOVATIVE INSTRUCTORS OF INFORMATION TECHNOLOGY

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

T. Case

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Carl J. Chimi
Bloomsburg University

Gene M. Gordon
Bloomsburg University

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

This paper examines the problems faced by instructors of information technology when they try to introduce new technologies for delivering instructional material. The authors have extensive experience in utilizing innovative instructional technologies, and are well aware of the rewards and frustrations encountered on the leading edge. Some of the issues discussed in this paper include:

1. *Dealing with new versions of software, when the institution wants to keep older versions on all of its computers*
2. *Dealing with budgets, both as constraints on resources and as inefficient mechanisms for providing leading edge technology*
3. *Dealing with colleagues, staff, and students who are not always in synch with leading edge needs*
4. *Conversely, we also have to deal with students whose own computers and software are sometimes ahead of our own*
5. *Using innovative technologies to deliver course materials – the rewards and the pitfalls*

INTRODUCTION

This paper is a step along the route that both authors have traveled for many years. It grew out of discussions among colleagues in our CIS department, which led into an eponymous symposium held before our student chapter of the Association for Information Technology Professionals (AITP) at our annual banquet for them. As such, it is the result of our experience and internal discussions, and is not meant as a quantitative research piece. It contains opinions and suggestions. Like most professors of Information Systems over the last two decades, we strive to provide our students with the most up-to-date, relevant, useful, and engaging education we can give within the constraints of time, budgets, university politics, technological change, shifting paradigms, and our own knowledge and endurance. Sometimes (not often), the path seems obvious. At other times, the path may seem obvious to us, but not to our colleagues. Sometimes our discipline shifts so radically as to make one wonder whether we have a real, sustainable discipline to teach.

The introduction of the Internet (to be precise, the making of a TCP-IP connection from our university out to the world) has greatly changed the manner in which we teach. For one thing, the Internet has made many people who might otherwise have been uninterested in computers take an interest; the possibilities for expression, artistic and entrepreneurial, are obvious to most people, as are the research and communications potential. We have had to

IR052394

BEST COPY AVAILABLE

recognize a changing set of needs among our primary stakeholders, including students and the businesses likely to employ them.

So we have tried to keep up and to change, while holding to the fundamentals of the Information Systems discipline. We have tried to be innovative in our methods of course delivery. We have faced many challenges and solved problems only to meet new challenges problems. The purpose of this paper is to discuss the issues we have faced walking along this path. In it, we will examine the problems faced by instructors of information technology when they try to introduce new technologies for delivering instructional material. We have extensive experience in utilizing innovative instructional technologies, and are well aware of the rewards and frustrations encountered on the leading edge.

BACKGROUND

The authors work at a state university, and have access to some advanced technologies. We have also, on our own, developed the use of a suite of applications, including web servers and ftp servers, and methodologies, such as delivering course materials over the World Wide Web. Our aim in trying to stay on the "leading edge" is to examine whether increased productivity can be expected from using new technologies. We would argue for an operative paradigm of shared decision-making. This approach can be a struggle, because in many institutions of higher learning and other organizations, an older model exists, in which managers who may have only a passing familiarity with new technologies make decisions about technology acquisitions and implementations. Most colleges and universities have administrative entities with titles such as "Academic Computing" and "Computer Services". These agencies generally administer the academic and administrative computers on the campus. While ostensibly service agencies, their purposes, decisions, and actions are often at odds with the needs of the innovative professor. These agencies can sometimes act as dictators of computing resources, thereby affecting technology-intensive curricula.

We argue in this paper that academic departments responsible for teaching technology-intensive curricula must have their own labs, under their direct control, in order to adequately present their courses. These departments are in a position directly analogous to science departments that require labs under their control to teach their curricula. Further, there must be a distinction made between production labs, which would be used for general day-to-day teaching, and experimental labs, which would be used for trying out new technologies without affecting the production labs.

As a final point of background, we will try to be careful to define what we mean when we use terms such as "leading edge", "innovative professors", and "technology-intensive curricula". We are using these terms in a narrow, specific way, and do not mean to imply that professors who do not use the same technologies, or who use other technologies in other ways, are not "innovative" or "leading edge". Fundamentally, we are talking about professors of Information Technology who, in some measure, try to eschew traditional methods of course delivery to incorporate technological delivery methods; we envision courses where the student not only cannot pass the course, but cannot perform many of the mundane aspects of the course without becoming intimately involved with the technology. Thus students do not receive paper syllabi; they must retrieve the syllabus from the Web. Thus, students never hand in assignments or tests on paper; all such submissions are handled electronically.

We have not struggled to create a taxonomy of issues to present in this paper. Rather, we have selected the ones that seem to us to be the most important, difficult, or challenging. The issues discussed include:

- Dealing with new versions of software, when the institution wants to keep older versions on all of its computers
- Dealing with budgets, both as constraints on resources and as inefficient mechanisms for providing leading edge technology
- Dealing with colleagues, staff, and students who are not always in synch with leading edge needs
- Conversely, we also have to deal with students whose own computers and software are sometimes ahead of our own
- Using innovative technologies to deliver course materials – the rewards and the pitfalls

DEALING WITH NEW VERSIONS OF SOFTWARE

It would be an interesting topic for other research to study which of the academic disciplines must deal with the most rapid changes in subject matter and course delivery materials. As Information Systems professors with over 40 years of combined experience, we believe a strong case

could be made for our field as experiencing the most rapid change. Although the fundamental topic areas (e.g. computer concepts, database concepts, systems analysis and design, etc.) have been recognized and set into the curriculum for many years, the actual technologies used to implement these concepts in the real world have undergone numerous radical changes over the years.

When we first came to our present positions in the early 1990s, our department was using DOS applications such as WordPerfect 5.1, dBaseIIIPlus, and Lotus 2.2. No attempt was being made to recognize that the Windows environment was beginning to emerge as the predominant microcomputer interface. It was a struggle for us to teach programs such as dBase, which had a copyright date of 1986, and to call ourselves up-to-date. Some of our colleagues argued that it didn't matter which software one used to teach the database concept, as long as one taught the concept clearly. This type of argument made it difficult to accept the argument made by the same colleagues that it would be impossible to use the Windows environment to teach, because the students would have trouble following the mouse, and because there was no command line interface.

Eventually, through forceful argument and simple market forces, we moved to a Windows 3.1 environment in our labs and classrooms. However many faculty strongly resisted that move.

Our University administration made the decision to provide a "common desktop" of applications across the computers of all staff, faculty, students, labs, and classrooms. Like all simplistic solutions to complex problems, this "common desktop" had many attractions and benefits to the campus as a whole. For example, it was now easier for the Academic Computing and Computer Services areas to administer individual software packages across the campus computer network, provided that a standard package for each type of application had been agreed upon.

Unfortunately, this has sometimes had very negative impacts upon our ability to stay current in our delivery of our courses. As IS professors, we have an obligation to develop our understandings of the latest ideas in software. This is at odds with the University's need to protect its investment in current technologies. Since the University has imposed (with reasonable and benign intent, it should be said) the common desktop, our ability to deliver the latest important software to our students has been compromised. One current example should make this issue clear.

When we finally moved to Windows95, the University standardized on the Microsoft Office95 suite of products to meet our corporate needs for word processing, spreadsheet processing, and database applications not served by our mainframe. At the time, in 1996, this coincided well with the changes in our department's curriculum, and our department was among the strongest advocates for bringing about this change. Many factions on campus were dead-set against making this change, including the secretarial staff, who had years of WordPerfect experience and understandably resisted the change to Word. By the Fall of 1996, our department had several up-to-date labs and classrooms with the "latest" software.

In the Spring of 1997, Microsoft introduced the Office97 suite of applications, which promised much better integration with the Internet, and were attractive adjuncts to our curriculum. We have still not moved as a University to this upgraded version of Office over a year and a half later, and we are not likely to do so for another year. The reasons are unclear, but they seem to be that the Director of Academic Computing is afraid that too many people will resist the upgrade. This points up the way in which a department that is heavily dependent on utilizing modern technologies can have its curriculum held hostage by corporate inertia.

One of the practical results of the above situation is that we and our students have had to deal with several different versions of software, because many of our students moved quickly to Office97, as did the authors of this paper. Although annoying and unnecessary, this is not a huge problem, except when using a product such as Microsoft Access, where the latest version (Access97) is not backwards compatible (can't be saved as) an earlier version (e.g. Access95 or Access 2). Microsoft's confusing version numbers for the earlier version (i.e., Access95 is also called version 7) have not helped this situation.

DEALING WITH BUDGETS

This is a simple issue. At many institutions, the budgeting process does not work well to serve the needs of Information Systems departments. First, budgets are set at least a year in advance, which leaves technology-dependent departments with the impossible task of predicting their needs well over as year in advance. Often, this leads to a continuation of the familiar, the same. Little room is left in the budgeting process for innovation. When innovations are introduced, they are often budgeted only for an initial year, and no provision

is made for ongoing, continuing expenses related to a particular innovation. This is not meant as an indictment of budgeting, nor of its practitioners. But there is a political and cultural difference between those who budget, who tend in that area of their lives to use conservative, sound fiscal practices based on past such practices, and academic innovators, who must remain open to changing winds within their fields; if we are correct that information technology fields change more rapidly than any other academic disciplines, then this leaves innovators within those fields as the ones most likely not to be adequately served by the budgeting process.

Fortunately, the news is not all gloomy on this front. Many organizations are starting to recognize the special needs of the technological innovator and are trying to accommodate. In our College of Business at our University, we have begun to recognize that some of our faculty are so experimental and innovative that their needs must be treated differently from the norm. We recognize these individuals by trying to upgrade their computers and software more often. And we expect them to keep innovating and experimenting.

DEALING WITH COLLEAGUES, STAFF, AND STUDENTS WHO ARE NOT ALWAYS IN SYNCH WITH LEADING EDGE NEEDS

We have touched in earlier sections on the tensions that exist between colleagues. Some of our colleagues in our own department argued against teaching using Windows because they felt it would be impossible. Many in the secretarial staff argued against changing word processors or spreadsheet programs because they already had an excellent command of WordPerfect and were convinced that Word lacked WordPerfect's capabilities and also that the learning curve would make many of them unproductive. The Accounting department, in particular, was heavily vested in using Lotus 1-2-3.

It has grown increasingly difficult to keep up with the pace of change in today's technologically driven world. Software companies have thus far demonstrated a remarkable capacity for marginal product upgrades that instill a need in consumers because of only one or two "killer features". The differences between the latest version of Microsoft Office (Office97 at the time of this writing) and Office two versions ago is actually very small. Almost any reasonably computer literate individual should be able to migrate from the earlier version to the later version and immediately have about 90% of the capability s/he had before upgrading. The other 10% will come very quickly, along with additional capabilities included in the new version. Thus upgrading from one version of software to another is generally no more daunting than learning to operate a new car. Yet people resist software upgrades.

In our field, this causes problems because there are compelling reasons for upgrading and always staying current. We list several below:

- Our students tend to stay current, either because they buy new computers with the latest software before entering college, or because they have a natural sense of what will be expected of them in college or their future workplaces and don't want to be left behind.
- Upgrading tends to standardize everyone at the same level, whereas not upgrading ensures that there will be a lack of standardization.
- Upgrading makes new "killer features" available. There are enough valuable features added into Office97 that did not exist in Office95 to make the argument that we should have upgraded over a year ago. By not upgrading, our students have been confused by different versions, and have either been denied these new capabilities, or have had to learn them on their own at their own expense.

Of course, the argument against immediate upgrading is that there are bound to be bugs in the new software. There is some truth to this argument, but since there are bugs in the old software, and there are always problems with computers, we feel that this argument does not have much force. We would argue that a department such as ours should never be more than one semester behind the introduction of a significant upgrade to a software product important to our curriculum.

DEALING WITH STUDENTS WHOSE OWN COMPUTERS AND SOFTWARE ARE SOMETIMES AHEAD OF OUR OWN

While this can be a very positive circumstance, it is often frustrating to have to deal with students who own later versions of software than those available on our university systems. This is not because we don't want students to utilize current technologies (see our feelings above), but because the different versions of software can cause incompatibilities and confusions among students and professors.

Students using newer versions of Office97 have already grown used to some of the newer convenient features (e.g. the Paste Function button in Excel, or the customizable toolbars of Office97, or the Visual BASIC for applications found in all Office97 products); it is hard to have to go back to less convenient ways of doing things found in earlier versions. Sometimes students have submitted assignments unknowingly in later versions of software formats, and professors who only have access to earlier versions are unable to read the assignments. Both authors of this paper have had to install, at their own expense, simultaneous versions of Office95 and Office97 in order to handle this type of situation.

Students also rightly feel that their professors should generally be more knowledgeable than most students about changes to important software packages. Not keeping up with the latest software can make a professor seem uninformed and ill equipped to teach, especially if the students are more current than the professor.

USING INNOVATIVE TECHNOLOGIES TO DELIVER COURSE MATERIALS

In this section we detail several of the techniques we have developed for weaving technology into the fabric of our courses. Techniques discussed include using the World Wide Web and ftp.

The World Wide Web – Online Syllabi

For each course we teach, we create a home page on our Web servers to which we create a link from our individual home pages. Each course home page consists of an HTML frame that has a list of options displayed on the left-hand area of the screen with the corresponding selection appearing to the right. The options usually include 1) a return to the professor's home page, 2) the course description, and 3) the course syllabus. Thus, if the student picks the course syllabus, the syllabus will be displayed in the right two-thirds of the screen.

The first day of each class, students are instructed to go to a computer lab, get into Netscape, and go to the instructor's home page. From there, students are shown how to find the appropriate course syllabus. Since no paper syllabus is handed out, students must use this resource. From the first day of classes, students are actively engaged in actually using the technology we are trying to teach them; the technology comes at them not only in the form of assignments, but also in the natural course of being students in our courses trying to get information about the courses.

This technique requires some knowledge of how to set up a web server, and of HTML. In our cases, since we each run our own web servers, it requires us to run our office machines 24 hours a day, seven days a week. If the machines are unreliable, which has occasionally happened, this will not work. It has also happened that a student working for Academic Computing came into one of the author's office uninvited and installed software that caused the author's computer to crash regularly each night.

FTP – Delivering Assignments

Similarly, when students have assignments or papers to hand in, they no longer hand in disks nor hard copy. All work is handed in electronically over the campus TCP/IP system using the File Transfer Protocol (ftp). FTP is a service that allows files to be copied from on machine to another over a TCP/IP network, such as the Internet.

Each of the authors has set up an ftp server on his office machine, which runs 24 hours a day under normal circumstances. The FTP server software chosen is called Serv-U, which is a shareware package; for a fee of about $20.00, one can get a very robust FTP server with an easy to understand interface. Using client ftp software available in all computer labs and in almost all home Internet setups, students can transmit their files at any time to the instructor's machine, using accounts set up specifically for each assignment.

Each account designates the course, the assignment, and the appropriate subdirectory for the assignment, so each assignment ends up exactly where we want it on our hard drives, ready to be graded. We have hit upon a system for creating User IDs and passwords for each assignment that has worked well for us. For example, if the course is 92.150 (Introduction to CIS) and the assignment is the first Excel assignment, the User ID would be 92.150.excel1. The password for the account is always the same as whatever comes after the last period in the User ID, therefore the password in this case would be excel1. Creating the account lists by assignment, rather than by individual student, saves us a lot of time and effort. However, it does mean that the students must have a system for naming their files, because duplicate filenames cannot exist. Our system is to require students to name the files they submit by their last names and assignment number. Under this system, Carl Chimi's first Excel assignment file would be called chimi1.xls. In the event of duplicate last names, the initial of the first name, or some other convenient discriminator, is used, e.g. chimic1.xls (Carl Chimi) and chimij1.xls (Jeanine Chimi).

Also, if a student submits a file, but then wishes to revise it, s/he must rename the revised file using a letter, e.g. chimic1b.xls. Only the latest version of the file is used for grading purposes.

Accounts are turned off at the designated due time, so late submissions are not possible through ftp. This system has many advantages, not the least of which is that, if performed correctly, it forces the student to have a backup copy of his or her assignment. We know of colleagues who encourage file submissions via attachments to email messages, and we teach this technique in classes where it is appropriate, but we feel that the FTP method is superior for general assignment delivery, because it puts the files right in the subdirectories on our hard drives where they belong. Email attachments have to be unattached and manually placed where they belong.

The World Wide Web – Online Exams

Examinations are rarely given on paper in our classes anymore. For each exam, an electronic template is created using a shareware product called Webforms. This automatically sets up an HTML copy of the exam but, more importantly, Webforms also contains functions for collecting the data for each student's exam into a Microsoft Access database, whence it can easily be graded. While this process can be time consuming on the front end for the instructor, it has many advantages. The important point is that students have to use the technology in order to take the exam (and they all learn to do it very quickly because they have to).

Webforms, like Serv-U, is a shareware package requiring a nominal registration fee (about $30.00). The interface can be a little tricky, but a familiarity with creating forms in HTML will make it easier to work with. Using Webforms, one creates the basic structure of the exam, i.e. all of the fields to be filled in (e.g. name and student number, text areas for essay questions, radio buttons for True/False or Multiple Choice questions), text to be displayed (questions, instructions, etc.), email address where the exam answers are to be sent, and submission and reset buttons. Once the structure is complete, Webforms will automatically generate the proper HTML code for the exam. Our experience is that Webforms, while very good, will not generate a completely useable HTML test; usually another HTML editor such as Netscape Communicator or the Windows Notepad is used to finalize the look of the exam.

The HTML exam file is placed on a University server that runs a World Wide Web server. At exam time, the students are given the URL for the exam and instructed to use Netscape to locate it. The students are also instructed (usually during the class period before the first exam) that their answers will be sent to the instructor in the form of an email message and that, therefore, each student must know how to set up Netscape to send email. Learning this procedure is considered to be part of the test; students are told that they must know it when they take the exam, no instruction in setting up Netscape will be given on exam day. Very few students come unprepared.

When a student is finished with the exam, s/he presses a Submit button, which sends the exam answers as an email message to the instructor's email account on a University UNIX server. The instructor monitors the server, and as each exam arrives, that student is notified. At that point the student is free to close Netscape and to leave. Students should not close the exam until they know for certain that the exam has arrived in the instructor's account. Students are also free to print their exams, if they desire.

Once all of the exams have arrived in the instructor's email account, Webforms comes into play again. The professional edition of Webforms allows the user to designate a POP3 mail server (such as the UNIX mail server mentioned above). Webforms will then go out to that server, examine all of the email messages found on it, and download each message which was generated by a file (such as the HTML exam) generated by Webforms. Essentially, it downloads each exam into its internal database. Form there, Webforms can export the data out to either a text file or a Microsoft Access database with each question in its own field.

With the data in Access, it is easy to automate the grading of True/False and Multiple Choice questions; essay questions must, of course, still be graded manually. The grades are recorded in an Excel spreadsheet and pasted onto an article posted to the class newsgroup, discussed below.

Students seem to really prefer this method of taking exams to using pen and paper, and they have been vocal in their preference. Somehow, once they are used to taking exams this way, the process seems to flow very quickly and smoothly. As instructors, we generally prefer to give exams this way now, but we warn our readers that there is more work involved, especially on the front end, in giving tests this way. The learning curve is substantial, but not insuperable, and the whole process is more complex than just printing and duplicating an exam. Having a knowledgeable graduate assistant who can do

some of the grunt work involved can ease some of the pain involved, as can the knowledge that your students are doing the very things you want them to learn in order to do something as mundane as taking a test.

CONCLUSION

While the authors do not have solutions to all of the problems to be presented, it is hoped that the discussions engendered by our paper will lead to a cross-fertilization of ideas between institutions. We envision this paper as the beginning of a dialogue among people who want to be innovative, but to be so in an informed and aware manner.